maintained in the device object, so any subsequent requests to the particular device object only require a checking of a bit rather than the entire registry.

If the current device object list does not locate the assigned LUN, a request to find new disk devices is sent through the I/O device control interface, i.e., IOCTL_DISK_FIND_NEW_DEVICES. The IOCTL_DISK_FIND_NEW_DEVICES request determines whether another device that the driver supports has been connected to the I/O bus, either since the system was booted or since the driver last processed this request. If such a device is found, the driver sets up any necessary system objects and resources to handle I/O requests for its new device. It also initializes the device on receipt of this request dynamically (i.e., without reboot). Such a driver is assumed to support devices connected on a dynamically configurable I/O bus.

This request generates a claim device to all the unassigned disks behind a particular port. The filter driver 354 then prevents any claiming of device objects that have yet to be assigned by intercepting the claim of each LUN, and comparing each LUN with devices available on that port. If the LUN exists, they are made available and the filter driver 354 makes the device objects available. Once a LUN is assigned, the operating system (e.g., Windows NT) maintains the device object for the course of the system up time. Therefore, the port driver 356 prevents an inordinate amount of device objects from being created during boot. If a disk device object cannot be claimed, it does not generate a device object. But if the LUN is found, the claim is successful, and a device object is created so that it can also then be checked to see if it should be masked or unmasked.

To unassign a LUN, the common storage area (e.g., Windows NT registry) is updated by removing that device's identification. A unique IOCTL is then sent to a filter driver (not shown) disposed "above" the SCSI class driver 352 to remove access to all device objects for the LUN that is to be unassigned. When unassigning a previously assigned LUN, only that filter driver needs to be notified because its device objects have already been created. This requires the submission of the IOCTL dedicated to that filter driver through the device I/O control API. Once the IOCTL is received, the disk id is checked against the registry, and if it no longer exists, the device object is masked from future I/O activity. If the unassigned LUN is later reassigned, the same filter driver, again, only needs to be notified (again, because the corresponding device objects already exists).

LUN Masking on Windows 2000 Hosts

In an embodiment of the invention for a Windows 2000 operating system, LUN masking is performed on hosts 12 in a manner similar to that described above with respect to a host running the Windows NT operating system.

The illustrated portion of the modified operating system 350 for a Windows™ host is architected and operated similarly to that described above with respect to the Windows™ NT operating system. LUN masking is performed in a similar fashion to that described above for Windows™ NT, except that the filter driver 354 intercepts the class driver 352 claims to storage devices (that are not assigned to the selected host 12), by blocking the claim requests generated by the class driver 352 in the first instance, rather than by blocking responses by the port driver 356 to the

class driver in response to such requests. As above, the blocking of claims requests in the Windows™ environment also prevents the class driver 352 from creating device objects, thereby, preventing the file system (or other aspects of the operating system or any applications program executing thereon) from accessing unassigned devices.

According to the illustrated embodiment, the agent 40 prevents masked LUNs from appearing in the Device Manger of the Windows™ 2000 interface by setting a flag in the data structure normally sent by the plug-and-play manager (not illustrated) with the device state query. In addition, the illustrated embodiment prevents the plug-and-play manager from generating notifications to an operator of a host 12 from which a masked device has been removed. This is accomplished by setting a flag in the data structure normally sent by the plug-and-play manager along with the device capabilities query.

In an alternate embodiment for a Windows™ 2000 host, masking is accomplished by modifying a data structure populated by the port driver to reflect LUNs (or other devices) that are attached to the host.

In normal operation of a Windows™ 2000 host, the plug-and-play manager (which is a conventional component of the Windows 2000 operating system) is initiated at boot-up and creates a data structure that it passes to the SCSI port driver 356. The port driver 356 populates that data structure with information regarding all found devices (e.g., SCSI addresses). The illustrated embodiment effects masking via the filter driver 354, which removes from that data structure information regarding fiber channel devices not listed in the table 354a. As a result,